****

Class: SY-MCA   Shift / Div: A   Roll Number: 52069

Name: YASH KANAPADE   Assignment No: 3   Date of Implementation: 03/11/23

****

Q1. Write black box test cases using ECP & BVA to test the requirement –

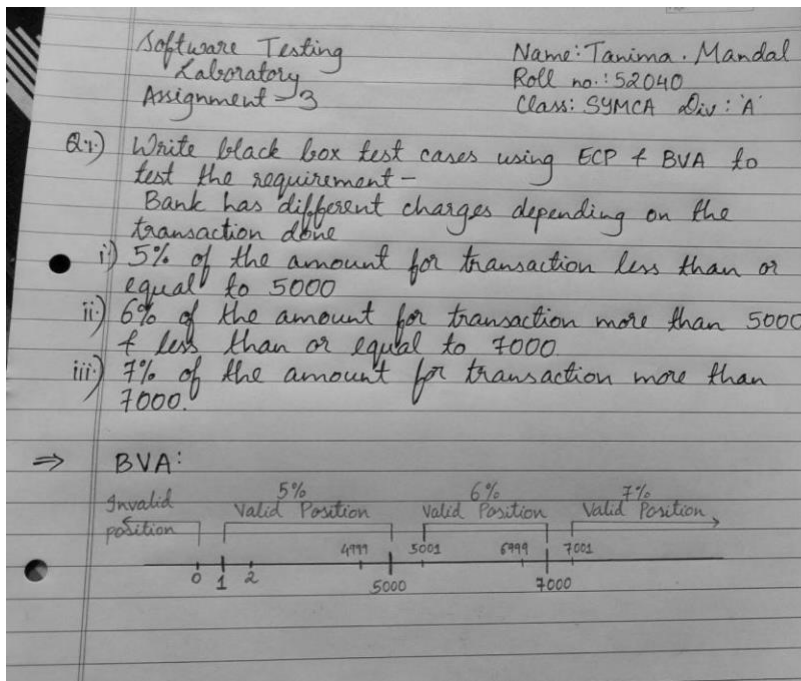Bank has different charges depending on the transaction done

5% of the amount for transaction less than or equal to 5000

6% of the amount for transaction more than 5000 and less than or equal to 7000

7% of the amount for transaction more than 7000

Solution:

BVA:

ECP:

Test Case Id	Partition	Type of input	Test Data	Expected output
BR_1	transaction amt. <=5000	Valid	1, 2, 4999, 5000	Transaction charges=5% of amt.
BR_2	transaction amt. >=5000 & <=7000	Valid	5001, 6999, 7000	Transaction charges=6% of amt.
BR_3	transaction amt. >7000	Valid	7001	Transaction charges=7% of amt.
BR_4	transaction amt. <1	Invalid	0	Error message "Invalid amt."

Q2. Write black box test cases using state based (state transition) technique to test the requirement –

Withdrawal of money from ATM. 'User A' wants to withdraw 30,000 from an ATM. Imagine he could take 10,000 per transaction and the total balance available in the account is 25,000. In the first two attempts, he could withdraw money. Whereas in the third attempt, the ATM shows a message as "Insufficient balance, contact Bank". Same Action but due to change in the state, he couldn't withdraw the money in the third transaction.

Solution:
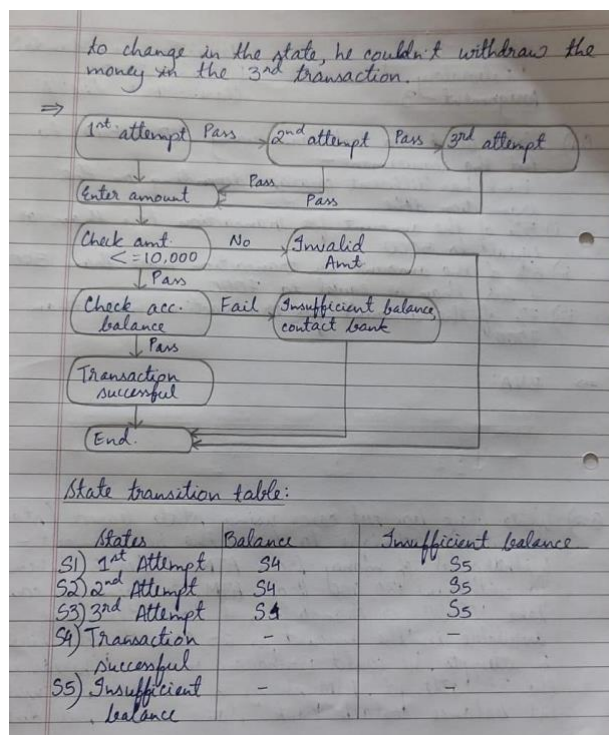
Test cases:

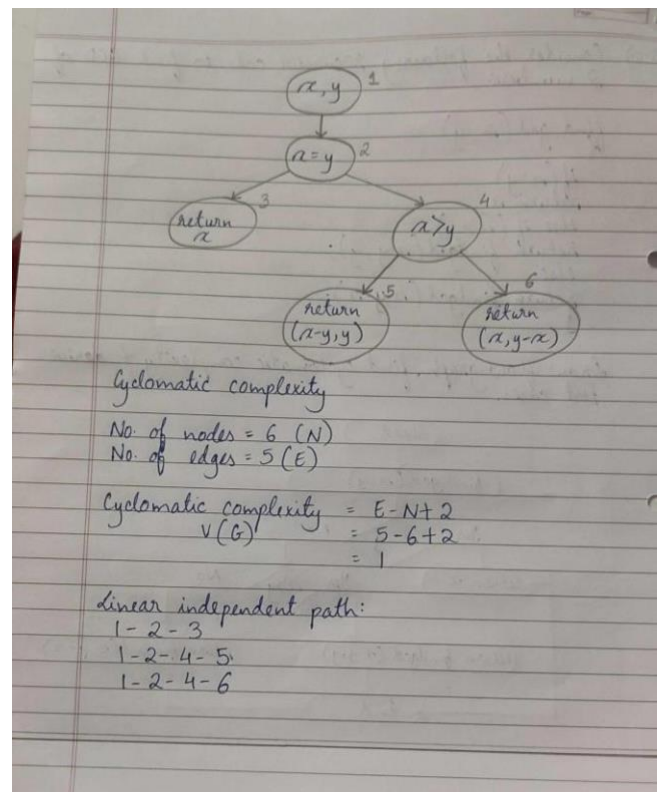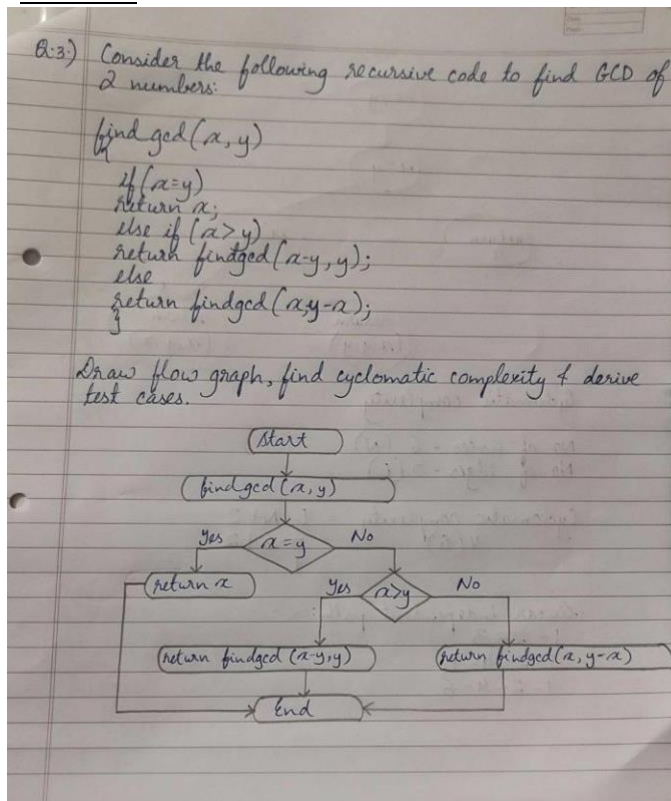| Test Case Id	Test scenario	Test step	Test data	Expected output	Actual output
ATM_1	Check withdraw money transaction amt.	withdraw Rs.10000	10000	Avail. balance=15000	Avail. balance=15000
ATM_2	Check withdraw money transaction amt.	withdraw Rs.10000	10000	Avail. balance=5000	Avail. balance=5000
ATM_3	Check withdraw money transaction amt.	withdraw Rs.10000	10000	Error message:"Insufficient balance,contact bank"	Error message:"Insufficient balance,contact bank"

Q3. Consider the following recursive code to find GCD of two numbers
find gcd (x , y)
{ if (x = y) return x;
else if (x>y) return
findgcd(x-y,y); else
return findgcd(x,y-x);
}
Draw flow graph, find cyclomatic complexity and derive test cases

Solution:

Q:3) Consider the following recursive code to find GCD of
2 numbers:

find gcd (x, y)
if (x=y)
Return x;
else if (x>y)
return findgcd(x-y,y);
else
return findgcd(x,y-x);

Draw flow graph, find cyclomatic complexity & derive test cases.

```
            Start
              |
        find gcd (x, y)
              |
    Yes      x=y      No
     |                 |
  return x    Yes    x>y    No
     |          |            |
          return findgcd(x-y,y)   return findgcd(x, y-x)
                    |
                   End
```

```
                    (x,y) 1
                      |
                    (x=y) 2
                   /      \
                3 /        \ 4
            (return        (x>y)
               x)          /    \
                        5 /      \ 6
                    (return    (return
                    (x-y,y))   (x,y-x))
```

Cyclomatic complexity

No. of nodes = 6 (N)
No. of edges = 5 (E)

Cyclomatic complexity = E - N + 2
        V(G)   = 5 - 6 + 2
                       = 1

Linear independent path:
1 - 2 - 3
1 - 2 - 4 - 5.
1 - 2 - 4 - 6

Test cases:

Test Case Id	Test Case Description	Steps	Test data	Expected output	Actual output
TC_001	find_gcd(x,y)	Read x and y such that x=y	x=5, y=5	return x	return x
TC_002	find_gcd(x,y)	Read x and y such that x>y	x=8, y=4	return find_gcd(x,y-x)	return find_gcd(x,y-x)
TC_003	find_gcd(x,y)	Read x and y such that x<y	x=3, y=9	return find_gcd(xy,y)	return find_gcd(x-y,y)